On the Equivalence of Sudan-Decoding and Decoding via Virtual Extension to an Interleaved Reed-Solomon Code

Alexander Zeh, Sabine Kampf, Martin Bossert

On the Equivalence of Sudan-Decoding and Decoding via Virtual Extension to an Interleaved Reed-Solomon Code

Alexander Zeh, Sabine Kampf and Martin Bossert
Department of Telecommunications and Applied Information Theory
University of Ulm, Germany
{alexander.zeh,sabine.kampf,martin.bossert}@uni-ulm.de

***Abstract*—In this paper we investigate two new decoding schemes for Reed-Solomon codes, which allow to decode beyond half the minimum distance. One is Sudan's list-decoding principle, based on interpolation with a degree-restricted bivariate polynomial. We show a syndrome-based approach of it. We compare Sudan's procedure with a scheme that is based on an extension to Interleaved Reed-Solomon codes. We present theoretical parallels and outline both algorithms in a unique comparable way. Furthermore, we show the connection of both schemes to the classical Linear Feedback Shift Register analysis. Afterwards, we compare the performance of the considered schemes.**

***Index Terms*—Interleaved Reed-Solomon (IRS) codes, Sudan interpolation, Multi-Sequence/Multi-Level Shift Register, Fundamental Iterative Algorithm (FIA), Berlekamp-Massey Algorithm (BMA)**

I. Introduction

Guruswami and Sudan [1], [2] found the first non-exponential-time list-decoding algorithms for Reed-Solomon (RS) codes in 1997 respectively 1999. They consist of an interpolation step and a factorization step of bivariate polynomials. While the work was focused on the existence of such an polynomial-time algorithm, an efficient implementation is in the focus of many researchers.

Recently Sudan's and Guruswami-Sudan's approach were reformulated to a univariate problem ([3], [4], [5]). In this contribution we consider Sudan's original approach which is applicable to RS codes with rate $R \leq 1/3$. The algorithm of Schmidt *et al.* [6], [7] is based on a virtual extension to Interleaved Reed-Solomon (IRS) codes and also allows an increase of the decoding radius for low-rate RS codes only. In fact, the rate-restriction is the same.

Recently, for both schemes a syndrome-based decoding method was derived [3], [6]. We compare them and their decoding algorithm, which is, for both schemes, an extension of the Fundamental Iterative Algorithm (FIA) of Feng and Tzeng [8]. The FIA itself can be seen as a generalization of the well-known Berlekamp-Massey Algorithm (BMA).

In the next section we introduce basic notations and the assumption under which we can compare both schemes. In Section III we outline the basic idea of virtual extension to an Interleaved Reed-Solomon (IRS) code, this decoding approach will be called IRS scheme throughout the paper. The syndrome-based Sudan decoding approach (or Sudan scheme) is explained in Section IV. Both algorithms and their connection to the classical BMA are presented in Section V. We compare their performance analytically and with some simulations in Section VI. Section VII concludes this contribution.

II. Principal Idea of the Comparison

A. Notation

By $\mathcal{RS}(n,k,d)$ a (generalized) Reed-Solomon code over a field $F = GF(q)$ with $n < q$ is denoted and given by

$$\mathcal{RS}(n,k,d) = \{\mathbf{c} = (f(\alpha_1), \ldots, f(\alpha_n)) : \deg f(x) < k\}, \tag{1}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_n$ are distinct nonzero elements of F (the code-locators). RS codes are in the class of MDS-codes and the minimum distance is given by $d = n - k + 1$. In the classical decoding process the received vector $\mathbf{r} = \mathbf{c} + \mathbf{e} = (r_1, r_2, \ldots, r_n)$ can contain up to $\tau_0 = \lfloor (n-k)/2 \rfloor$ errors. The core of the decoding process is solving the classical key equation:

$$S(x) \cdot \Lambda(x) \equiv \Omega(x) \mod x^{n-k}, \tag{2}$$

where the degree of the error-locator polynomial $\Lambda(x)$ is here denoted by τ_0. The so-called error-evaluator polynmial $\Omega(x)$ satisfies $\deg \Omega(x) < \tau_0$. The key equation can be solved using the BMA that performs linear-feedback shift-register synthesis or by the extended Euclidean Algorithm.

B. Assumptions for the Comparability of both schemes

The decoding result of the decoder based on a virtual extension to an IRS code returns either a unique result, where the returned error-locator gives us the location of the errors (increased decoding radius τ), or the decoder declares a decoding failure. In contrast to this a list-decoder returns a list (where the maximum number of possible codewords is limited to l) with all codewords of maximal distance τ to the received word. Note, that the Sudan approach [1] guarantees that the sent codeword is always on the list. For comparability we "simplify" our list-decoder. It declares a decoding failure if the outputted list contains more than one codeword.

ISBN 978-3-8007-3211-1

III. IRS Scheme

A. Principal Idea

The decoding with the IRS scheme relies on having l errors that occurred in the same positions yet have independent error values. Hence there exist l key equations:

$$S^{(t)}(x) \cdot \Lambda(x) \equiv \Omega^{(t)}(x) \mod x^{n-k}, \forall t = 1, \ldots, l \quad (3)$$

which are all solved by the same error-locator polynomial $\Lambda(x)$. So compared to the case of classical decoding the number of equations available for the determination of $\Lambda(x)$ is increased allowing for decoding beyond half the minimum distance. The system of equations now takes the following form

$$\left(\mathcal{S}^{(1)}\ \mathcal{S}^{(2)}\ \cdots\ \mathcal{S}^{(l)}\right)^T \cdot \mathbf{\Lambda} = \mathbf{0}. \quad (4)$$

We denote by $\mathbf{S}_{IRS} = (\mathcal{S}^{(1)}\mathcal{S}^{(2)}\cdots\mathcal{S}^{(l)})^T$. Note, each matrix $\mathcal{S}^{(i)}$ has the form of a Hankel matrix:

$$\begin{pmatrix} \mathcal{S}_0^{(i)} & \mathcal{S}_1^{(i)} & \mathcal{S}_2^{(i)} & \cdots & \mathcal{S}_{wt(\mathbf{e})}^{(i)} \\ \mathcal{S}_1^{(i)} & \mathcal{S}_2^{(i)} & \cdots & & \mathcal{S}_{wt(\mathbf{e})+1}^{(i)} \\ \vdots & & & & \vdots \\ \mathcal{S}_{d_i-wt(\mathbf{e})}^{(i)} & \cdots & & & \mathcal{S}_{d_i}^{(i)} \end{pmatrix}, \quad (5)$$

where d_i is the degree of the i-th syndrome polynomial $S^{(i)}(x)$ and $wt(\mathbf{e})$ the number of errors. As the name indicates, this decoding approach was originally developed for IRS code, and in order to apply it to ordinary RS codes, it is necessary to construct additional syndromes from the received word. One method to do this has first been described in [6]: The received word is raised element-wise to the tth power ($t = 2, \ldots, l$). The resulting codeword and error will be described exemplary in the following for the case of element-wise squaring, yet the result can easily be extended to higher powers. By element-wise squaring $\mathbf{r}$ one obtains

$$\mathbf{r}^{[2]} = \left(r_1^2,\ r_2^2,\ \ldots\ , r_n^2\right) \quad (6)$$

where

$$r_i^2 = (c_i + e_i)^2 = c_i^2 + 2c_ie_i + e_i^2. \quad (7)$$

We interpret $\mathbf{r}^{[2]}$ as the sum of the new codeword $\mathbf{c}^{[2]}$ and the new error word $\mathbf{e}^{[2]}$, so we have

$$\mathbf{c}^{[2]} = \left(c_1^2,\ c_2^2,\ \ldots\ , c_n^2\right) \quad (8)$$

and

$$\mathbf{e}^{[2]} = \left(2c_1e_1+e_1^2,\ 2c_2e_2+e_2^2,\ \ldots\ , 2c_ne_n+e_n^2\right). \quad (9)$$

Due to the element-wise operation, the errors cannot propagate and hence $\mathbf{e}^{[2]}$ is (generally) an error with the same weight as $\mathbf{e}$ at the same error positions, yet linearly independent[1].

On the other hand, $\mathbf{c}^{[2]}$ is a codeword of a $\mathcal{RS}(n, k^{[2]} = 2k-1, d)$ code: Each element of $\mathbf{c}^{[2]}$ can be written as $c_i^{[2]} = c_i^2 = f(\alpha_i)^2$, hence it is the evaluation of the polynomial $f^{[2]}(x) = f^2(x)$ at the same code locators used for the original RS code. From $\deg f^{[2]}(x) \leq 2(k-1)$ the dimension of the code follows.

Continuing this reasoning, one finds that when raising the received word to the tth power it is possible to write the result as sum of a (generally) independent error word and the codeword of a $\mathcal{RS}(n, k^{[t]} = t(k-1)+1, d)$ code. Consequently, the degree of the tth syndrome polynomial is

$$\deg S^{(t)}(x) < n - k^{[t]} = n - t(k-1) - 1 \quad (10)$$

according to the classical syndrome definition. The tth syndrome contributes $n - t(k-1) - wt(\mathbf{e})$ equations for the determination of the error locator polynomial. Using the syndromes $S(x), S^{(2)}(x), \ldots, S^{(l)}(x)$ it is possible to correct up to (see [7]):

$$\tau = \left\lfloor \frac{2ln - l(l+1)k + l(l-1)}{2(l+1)} \right\rfloor \quad (11)$$

errors.

B. Homogeneous Set of Equations as Multi-Sequence Shift Register Problem

The set of homogeneous equations (4) is a Multi-Sequence Linear Shift Register Problem. Figure 1 illustrates this problem. It is an extension of the classical BMA algorithm, where only one sequence occurs. For details see [9].

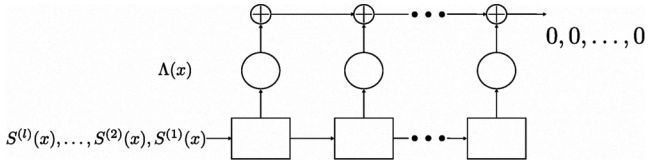

Fig. 1. IRS Scheme as Multi-Sequence Shift Register Problem

IV. Sudan-Scheme

A. Reformulation of the interpolation problem to a key equation

Sudan's original list-decoding algorithm ([1]) was reformulated by Roth and Ruckenstein [3], [4] to a key equation which is an extension of the classical key equation for half-minimum distance decoding. We directly present this equation in the following (for the derivation the reader is referred to [3], [4], [10]). The focus of this section is set to the Berlekamp-Massey [11] like algorithm solving the resulting set of homogeneous equations for the Sudan principle efficiently. In the Sudan decoding procedure for RS codes, we search a bivariate polynomial $Q(x, y) = \sum_{t=0}^{l} Q^{(t)}(x)y^t$, where $Q(\alpha_i, r_i) = 0 \forall i = 1, \ldots, n$. Furthermore, the degree of $Q^{(t)}(x)$ is smaller than $N_t = n - \tau - t(k-1)$. The y-roots (at most l) give us the possible sent codewords. Let $\Lambda^{(t)}(x)$ denote the reciprocal polynomials of $Q^{(t)}(x)$. Then the reduced set (without $Q^{(0)}(x)/\Lambda^{(0)}(x)$) can be written as (for details see [3]):

$$\sum_{t=1}^{l} \Lambda^{(t)}(x) \cdot x^{(t-1)(k-1)} \cdot S^{(t)}(x) \equiv \Omega(x) \bmod x^{n-k}, \quad (12)$$

[1] Due to the properties of finite fields there is a certain probability that the new error word is of less weight or linearly dependent to $\mathbf{e}$.

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach

where

$$\deg \Omega(x) < n - k - \tau. \tag{13}$$

Furthermore the l Sudan syndrome polynomials $S^{(t)}(x)$ are the first terms of the well-defined formal power series $S_{\infty}^{(t)}(x)$ defined as:

$$\frac{\overline{R}(x)^t}{G(x)} = x^{(t-1)(n-1)} \cdot S_{\infty}^{(t)}(x) + U^{(t)}(x), \tag{14}$$

where $R(x)$ is the Lagrange interpolation polynomial, s.t. $R(\alpha_i) = r_i \ \forall i = 1, \ldots, n$, and $\overline{R}(x)$ is its reciprocal counterpart. The polynomial $G(x)$ is $G(x) = \prod_{i=1}^{n}(1 - \alpha_i x)$. We emphasize that the syndromes are the same than for the scheme discussed in Section III. Clearly, for $l = 1$ Equation (12) becomes (2).
The resulting set of τ homogeneous equations (we consider the terms of (12) with the highest degree) can be written in matrix form:

$$\left(\mathcal{S}^{(1)} \quad \mathcal{S}^{(2)} \quad \cdots \quad \mathcal{S}^{(l)}\right) \cdot \mathbf{Q}^* = \mathbf{0}, \tag{15}$$

where $\mathbf{Q}^* = (\mathcal{Q}^{(1)}\mathcal{Q}^{(2)} \cdots \mathcal{Q}^{(l)})^T$ and $\mathcal{Q}^{(t)} = (Q_0^{(t)}, Q_1^{(t)}, \ldots, Q_{N_t-1}^{(t)})^T$ is the vector representation of the bivariate interpolation polynomial in the shortened form $Q^*(x, y) = \sum_{t=1}^{l} Q^{(t)}(x) y^t$.

The missing $Q^{(0)}(x)$ can be interpolated with $N_0 = n - \tau$ pairs (α_i, r_i) because of the relation:

$$Q^{(0)}(\alpha_i) = -Q^*(\alpha_i, r_i) = -\sum_{t=1}^{l} Q^{(t)}(\alpha_i) y_i^t, \quad i = 1, \ldots, n.$$

Note, that $\mathbf{S}_{Sudan} = (\mathcal{S}^{(1)}\mathcal{S}^{(2)} \cdots \mathcal{S}^{(l)})$ is a $\tau \times \sum_{t=1}^{l} N_t$ matrix.

B. Homogeneous Set of Equations as Multi-Level Shift Register Problem

Similar to the IRS-based scheme we can represent the reformulated Sudan interpolation problem in terms of a Linear Feedback Shift Register problem. In contrast to the IRS-scheme the syndrome polynomials and the corresponding polynomials $\Lambda^{(t)}(x)$ form a Multi-Level Shift Register as shown in Figure 2. We point out that we search here l different polynomials $\Lambda^{(t)}(x)$ (with different degree) and that the sum of the linear combination should be zero. Now, we investigate both problems and show algorithms solving them.

V. Parallels and Differences of both Schemes

A. On the parameters

As already mentioned the syndrome polynomials $S^{(t)}(x)$ are the same for both considered schemes. Also the parameter l and the increased decoding radius τ (see (11)) are equal and depend on the code length n and its dimension k. Nevertheless the basic ideas and the resulting sets of homogeneous equations are different. Both schemes can be reduced (for $l = 1$) to the classical case (see (2)). Both algorithms are an extension of the BMA and we will compare them in the following.

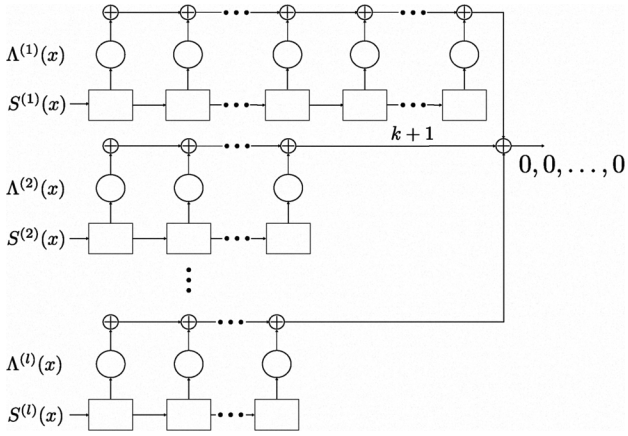

Fig. 2. Sudan's interpolation constraints as Multi-Level Shift Register Problem

B. Unique presentation of both algorithms

The Fundamental Iterative Algorithm (FIA) of Feng-Tzeng [8] finds the minimal number of first columns of an arbitrary matrix which are linearly dependent. It is well-known that when the FIA is tailored to a Hankel matrix (such as in (5)) it coincides with the BMA. In this section we will show an extension of the FIA for both the homogeneous set of equations coming from the IRS-scheme and the reformulated Sudan interpolation constraints. In the IRS-scheme l Hankel matrices are arranged vertically (see (4)), while for Sudan the syndrome matrices are arranged horizontally (see (15)).
In both schemes, the rows respective the columns of the syndrome matrices $\mathbf{S}_{IRS}$/$\mathbf{S}_{Sudan}$ will be interchanged in a similar manner. First we define the ordering $\prec$, which occurs in both algorithms. Let $\prec$ denote the order over the set of pairs $\{(i, t) | i \in \{1, \ldots, l\}, t \in \mathbb{N}\}$, where $(i, t) \prec (i', t')$ if and only if:

$$\begin{gathered} i + t(k-1) < i' + t'(k-1) \\ \text{or} \\ i + t(k-1) = i' + t'(k-1) \text{ and } t < t'. \end{gathered} \tag{16}$$

By $\succ_{\prec}$ (i, t) the pair that immediately follows (i, t) with respect to order defined by $\prec$ is denoted. Now we will describe both algorithms based on the FIA and show their connection to the classical Linear Feedback Shift Register (LFSR) analysis. Preliminary, let us redefine both problems with the help of the inner-product. The inner-product $\langle a(x), b(x) \rangle$ of two univariate polynomials $a(x) = \sum_{i=0}^{a} a_i x^i$ and $b(x) = \sum_{i=0}^{b} b_i x^i$ is defined as $\sum_{i=0}^{\min(a,b)} a_i b_i$. The inner-product $\langle a(x, y), b(x, y) \rangle$ for two bivariate polynomials $a(x, y) = \sum_{i=0}^{a_1} \sum_{j=0}^{a_2} a_{i,j} x^i y^j$ and $b(x, y) = \sum_{i=0}^{b_1} \sum_{j=0}^{b_2} b_{i,j} x^i y^j$ is defined as $\sum\sum a_{i,j} b_{i,j}$.

Problem 1 (Sudan scheme) *Let $S(x, y) = \sum_{t=1}^{l} S^{(t)}(x) y^t$ be the bivariate Sudan syndrome polynomial, where each $S^{(t)}(x)$ $\forall t = 1, \ldots, l$ as defined in (14). Then we search a nonzero bivariate polynomial $T(x, y)$ such that:*

$$\langle x^{\kappa} T(x, y), S(x, y) \rangle = 0 \quad \forall \kappa = 0, \ldots, \tau - 1, \tag{17}$$

holds.

ISBN 978-3-8007-3211-1

Note that $T(x,y)$ is the shortened Sudan interpolation polynomial $Q^*(x,y)$. The IRS scheme can be formulated as follows. Note that the dimension of the matrix $\mathbf{S}_{IRS}$ depends on the number of errors (see (5)). Let N_t^{IRS} denote the number of rows of the matrix $\mathcal{S}^{(t)}$ for the IRS-scheme. It is:

$$N_t^{IRS} = N_t - 1 + (\tau - wt(\mathbf{e})), \tag{18}$$

where τ is according to (11) the number of maximal correctable errors of the IRS/Sudan-scheme and $wt(\mathbf{e}) \leq \tau$ is the number of errors that really occurred.

Problem 2 (IRS scheme) *Let the l syndrome polynomials $S^{(t)}(x)$ $\forall t = 1,\ldots,l$ as defined in (14) and let N_ϑ^{IRS} be as defined in (18). Then for the IRS scheme we search a nonzero univariate polynomial $T(x)$ such that:*

$$\langle x^\kappa T(x), S^{(\vartheta)}(x)\rangle = 0 \; \forall \kappa = 0,\ldots,N_\vartheta^{IRS} - 1/\vartheta = 1,\ldots,l \tag{19}$$

holds.

Note that $T(x)$ is the reciprocal polynomial of extended error-locator polynomial with $0 < wt(\mathbf{e}) \leq \tau$ roots indicating the error locations.

C. Multi-Level Algorithm for Sudan

Algorithm 1 solves Problem 1 efficiently.

Algorithm 1: Algorithm for the Multi-Level Problem

Input: Biv. polynomials $S(x,y) = \sum_{t=1}^{l} S^{(t)}(x)y^t$;
Output: Bivariate polynomial $T(x,y)$;

Data: $T(x,y)$, Column pointer (ν,μ), Row pointer κ, Arrays $D[i]$, $A[i]$, $R[i]$, Variable Δ;

```
1  while κ < τ do
2      if compute then
3          Δ ← ⟨x^κ · T(x,y), S(x,y)⟩;
4      else
5          if R[ν] < 1 then
6              T(x,y) ← y^ν · x^μ;
7              Δ ← S_μ^(ν);
8              κ ← 0;
9          else
10             T(x,y) ← x · A[R[ν]](x,y);
11             Δ ← D[R[ν]];
12             κ ← R[ν] − 1;
13         end
14         compute ← TRUE;
15     end
16     if Δ = 0 or D[κ] ≠ 0 then
17         if Δ ≠ 0 then
18             T(x,y) ← T(x,y) − (Δ/D[κ]) · A[κ](x,y);
19         end
20         κ ← κ + 1;
21     else /* Δ ≠ 0 and D[κ] = 0 */
22         A[κ](x,y) ← T(x,y);
23         D[κ] ← Δ;
24         R[ν] ← κ;
25         compute ← FALSE;
26         (ν,μ) ←≻_≺ (ν,μ);
27     end
28 end
```

In contrast to the classical FIA, Algorithm 1 scans the l Hankel matrices $\mathcal{S}^{(i)}$ in parallel. The columns of $\mathcal{S}^{(i)}$ are virtually interchanged according to the $\prec$-ordering. The discrepancy calculation (Line 3) and the update rule (Line 18) are suited for bivariate polynomials. The discrepancy is stored in the array D and the intermediate polynomial in A. The row-pointer for every sub-matrix $\mathcal{S}^{(i)}$ is stored in the array R. Similar to the FIA for one Hankel matrix we can jump in each sub-matrix $\mathcal{S}^{(\vartheta)}$ to the previous row $\kappa - 1$ instead of row zero (see Line 12). This is the point where the complexity reduction comes from. For more details see [3], [10]. Without a proof we state that Algorithm 1 has time complexity $\mathcal{O}\left(\tau^2 l\right)$. Note that $\mathcal{O}\left(\tau^2\right)$ is the complexity for one $\tau \times (\tau + 1)$ Hankel matrix (classical decoding). We illustrate the functioning of Algorithm 1 in the following example.

D. Multi-Sequence Algorithm for the IRS-scheme

Algorithm 2 solves Problem 2 efficiently. We will describe the extension Algorithm 2 to the FIA tailored for one Hankel matrix in the following (for details see [8], [10]).

Algorithm 2: Algorithm for the Multi-Sequence Problem

Input: Univariate polynomials $S^{(t)}(x)$ $\forall\, t \in \{1,\ldots,l\}$
Output: Univariate polynomial $T(x)$;

Data: $T(x)$, Column pointer ψ, Row pointer (ϑ,κ), Row counter ρ, Arrays $D[i][j]$, $A[i][j]$, $R[i]$, Variable Δ;

```
1  while (ϑ,κ) < (l, N_ϑ − 1) do
2      if compute then
3          Δ ← ⟨x^κ · T(x), S^(ϑ)(x)⟩;
4      else
5          if κ < 1 and ϑ = 0 then
6              T(x) ← x^ψ;
7              Δ ← S_ψ^(ϑ);
8              (ϑ,κ) ← (1,0);
9          else
10             T(x) ← x · T(x);
11             if κ = 0 then
12                 (ϑ,κ) ← (ϑ − 1, k − 1);
13                 Δ ← 0;
14             end
15             κ ← κ − 1;
16         end
17         compute ← TRUE;
18     end
19     if Δ = 0 or D[ϑ][κ] ≠ 0 then
20         if Δ ≠ 0 then
21             T(x) ← T(x) − (Δ/D[ϑ][κ]) · A[ϑ][κ](x);
22         end
23         (ϑ,κ) ←≻_≺ (ϑ,κ);
24     else /* Δ ≠ 0 and D[ϑ][κ] = 0 */
25         A[ϑ][κ](x) ← T(x); D[ϑ][κ] ← Δ;
26         ψ ← ψ + 1;
27         compute ← FALSE;
28     end
29 end
```

The row pointer (ϑ,κ) is ordered with respect to (16) and used to index the two dimensional discrepancy array D. This stores (in contrast to the classical FIA) the discrepancy for each sub-

TABLE I
ORDERING $\prec$ FOR THE ROW/COLUMN POINTER

Column/Row of the syndrome matrices and the corresponding $\prec$-ordering					
0	(0,1)	7	(5,1)	14	(1,3)
1	(1,1)	8	(2,2)	15	(8,1)
2	(2,1)	9	(6,1)	16	(5,2)
3	(3,1)	10	(3,2)	17	(2,3)
4	(0,2)	11	(0,3)	18	(9,1)
5	(4,1)	12	(7,1)	19	(6,2)
6	(1,2)	13	(4,2)	20	(3,3)

matrix $\mathcal{S}^{(i)}$ $\forall i = 1, \ldots, l$. The array A stores the intermediate connection polynomial $T(x)$.
The discrepancy calculation in Line 3 and the update rule of the connection polynomial $T(x)$ (Line 21) are suited for Problem 2.
In the presentation of Algorithm 2 and the following example we assumed that $wt(\mathbf{e}) = \tau$ errors occurred. In general the rank of the matrix $\mathbf{S}_{IRS}$ should be calculated first and then the stop-condition (Line 1) has to be adjusted.

E. Example

We consider an $\mathcal{RS}(31, 4, 28)$ code over GF(31). From the parameter analysis we obtain a list size $l = 3$ and an increased decoding radius $\tau = 18$. The information polynomial $f(x) = \sum_{i=0}^{k-1} 1x^i$ is encoded according to Equation (1) and an error of weight 18 is added. We obtain the following values:

$$\begin{aligned}
\mathbf{c} &= (4, 9, 14, 11, 17, 20, 29, 12, 20, 26, 1, 24, 27, 10, 26, 0, \\
&\quad 11, 26, 23, 25, 1, 20, 10, 7, 15, 11, 6, 10, 28, 21, 1) \\
\mathbf{e} &= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, \\
&\quad 14, 15, 16, 17, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\
\mathbf{r} &= (5, 11, 17, 15, 22, 26, 5, 20, 29, 5, 12, 5, 9, 24, 10, 16, 28, \\
&\quad 13, 23, 25, 1, 20, 10, 7, 15, 11, 6, 10, 28, 21, 1). \qquad (20)
\end{aligned}$$

The corresponding $l = 3$ syndromes for both schemes (in vector notation) are:

$$\begin{aligned}
\mathcal{S}^{(1)} &= (15, 19, 18, 6, 8, 3, 1, 11, 17, 4, 7, 5, 18, 5, \\
&\quad 17, 9, 24, 15, 26, 9, 11, 8, 6, 24, 18, 15, 5) \\
\mathcal{S}^{(2)} &= (22, 0, 30, 8, 16, 26, 15, 22, 21, 28, 12, 27, 15, \\
&\quad 29, 5, 9, 13, 0, 2, 20, 27, 14, 1, 2) \\
\mathcal{S}^{(3)} &= (27, 25, 7, 12, 4, 2, 7, 5, 3, 0, 24, 26, 21, 23, 4, 24, \\
&\quad 1, 16, 15, 29, 14). \qquad (21)
\end{aligned}$$

The ordering according to $\prec$ as defined in (16) for the $\mathcal{RS}(31, 4, 28)$ code and our decoding schemes is listed in Table I. The Sudan syndrome matrix $\mathbf{S}_{Sudan}$ for the considered RS-code has 18 rows and 21 columns. For the IRS-scheme the corresponding syndrome matrix $\mathbf{S}_{IRS}$ is a 18×19 matrix. In Figure 3 the column pointer (μ, ν) for the syndrome matrix $\mathbf{S}_{Sudan}$ for the syndromes (21) is illustrated. The dots indicate the positions where in Algorithm 1 a non-zero discrepancy was calculated and no intermediate polynomial was stored before and so we can enter next column. Let us consider the point $(2, 2)$: Algorithm 1 examines column 3 of the first matrix $\mathcal{S}^{(1)}$ (column pointer is $(\nu = 1, \mu = 2)$) in the 3rd row. Then it

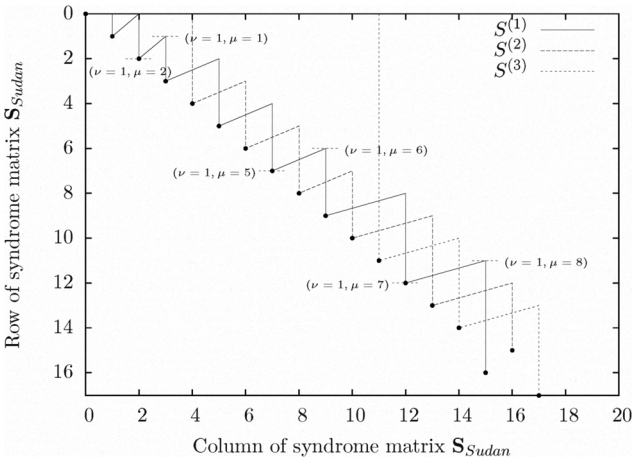

Fig. 3. Illustration of the discrepancy-calculation for the Sudan decoding procedure for an $\mathcal{RS}(31, 4)$ with $l = 3$ code.

enters column $(\nu = 1, \mu = 3)$ and can start in row 2 (Hankel property).
If Algorithm 1 examines the part of the matrix $\mathbf{S}_{Sudan}$ where the columns are interchanged, it "jumps" more than one column. See point $(\nu = 1, \mu = 5)$ to $(\nu = 1, \mu = 6)$, or $(\nu = 1, \mu = 7)$ to $(\nu = 1, \mu = 8)$ as indicated in Figure 3.
The discrepancy calculation of Algorithm 2 when applied to the 18×19 syndrome matrix $\mathbf{S}_{IRS}$ with the syndromes of (21) is illustrated in Figure 4. Here the rows are interchanged

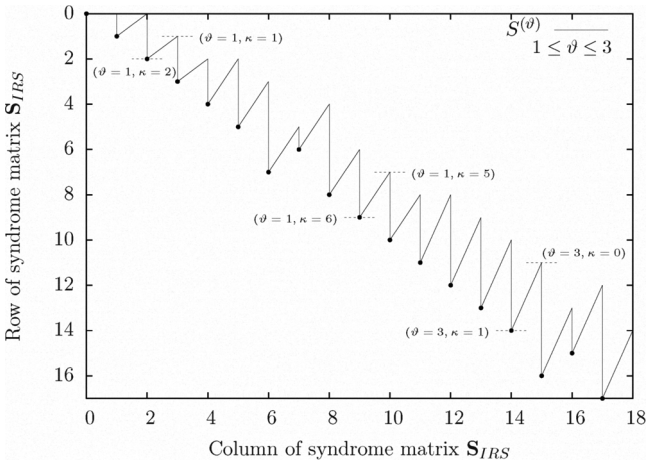

Fig. 4. Illustration of the discrepancy-calculation for IRS-scheme for an $\mathcal{RS}(31, 4)$ virtually extended to $l = 3$ Reed-Solomon codes.

in $\prec$-order. Let us consider the point $(14, 14)$ in Figure 4. Algorithm 2 calculates a nonzero discrepancy for the second row of the third sub-matrix $\mathcal{S}^{(3)}$ (indicated with the column pointer $(\vartheta = 3, \kappa = 1)$). Due to the Hankel property Algorithm 2 can start in the next column with the first row of $\mathcal{S}^{(3)}$ $((\vartheta = 3, \kappa = 0))$.
Algorithm 1 and Algorithm 2 take advantage of the Hankel structure in a similar manner and achieve a comparable time complexity.

ISBN 978-3-8007-3211-1

VI. Performance Analysis

A. Sudan-List-1 Decoder

In [12, Appendix D] several bounds for a Guruswami-Sudan list-decoder were derived. As the Sudan decoding procedure can be seen as a special case of GS, we can use them here, too. The considered bound is denoted by $\overline{L}_1(wt(\mathbf{e}), \tau)$, where $wt(\mathbf{e})$ is the number of erros occured and τ is the maximum decoding radius. $\overline{L}_1(wt(\mathbf{e}), \tau)$ gives us the probability for more than one codeword on the list and therefore is the probability that our modified list-decoder fails $P_{Sudan}(wt(\mathbf{e})) \lesssim \overline{L}_1(wt(\mathbf{e}), \tau) = \frac{1}{(q-1)^r} \sum_{s=d-wt(\mathbf{e})}^{\tau} (q-1)^s \left(\sum_{w=d-wt(\mathbf{e})}^{s} \binom{n-wt(\mathbf{e})}{w} \binom{wt(\mathbf{e})}{s-w} \right)$ (where $r = n-k$ and $d = n-k+1$).

Note that for a list-decoder the list can contain more than one codeword even if the number of errors is smaller than $\lfloor \frac{n-k}{2} \rfloor$. For both schemes the failure probability for a $\mathcal{RS}(255, 63, 193)$ code is plotted in Figure 5. For $l = 2$

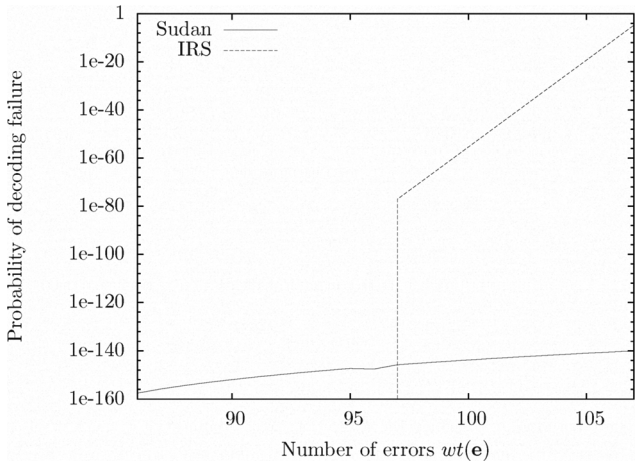

Fig. 5. Probability of decoding failure of both considered schemes for a $\mathcal{RS}(255, 63, 193)$ code.

we get from (11) for both schemes $\tau = \lfloor 107.33 \rfloor = 107$ (where $\tau_0 = 96$). The decoding failure probability for the IRS-scheme is zero, if the number of errors $wt(\mathbf{e}) \leq \tau_0$. Contrary a list-decoder can output several codewords even if $wt(\mathbf{e}) \leq \tau_0$. With our assumption the decoder fails in this case. Nevertheless even with this oversimplification the performance of the list-decoder is better (the complexity is similar), because the decoding failure probability is very low over the whole range of correctable error-weights.

B. Failure probability for IRS Scheme

An upper bound for the failure probability of an IRS decoder were derived in [7]. For virtual extension, this bound does not necessarily hold any more as the errors are no longer independent. However, simulations have shown a good coincidence of the actual failure probability to the upper bound derived. This probability of the IRS scheme given a certain number of errors $wt(\mathbf{e})$ can be approximated by $P_{IRS}(wt(\mathbf{e})) \lesssim \frac{1}{q-1} q^{-(l+1)(\tau - wt(\mathbf{e}))}$.

VII. Conclusion

We compared two decoding schemes that allow to decode beyond half the minimum distance. Both approaches are extensions of the classical Berlekamp-Massey approach. The increased decoding radius τ and the l syndromes are the same. Both schemes are comparable, but solve different problems: While the reformulated Sudan interpolation conditions lead to a Multi-Level Shift Register, the IRS-based scheme is a Multi-Sequence Shift Register problem.

Based on the Fundamental Iterative Algorithm an efficient implementation was presented and the complexity is similar. The case where the list of the Sudan-decoder contains more than one possible codeword on the list and the case when the IRS-based scheme fail do not coincide.

Note that our modified list-decoder fails when more than one codeword is on the list, but still outperforms the IRS-based scheme. It is more practical to choose the codeword on the list with the smallest hamming distance to the received word.

References

[1] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, 1997. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.1006/jcom.1997.0439

[2] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=782097

[3] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *Information Theory, IEEE Transactions on*, vol. 46, no. 1, pp. 246–257, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=817522

[4] G. Ruckenstein, "Error decoding strategies for algebraic codes," Ph.D. dissertation, Technion, 2001. [Online]. Available: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2001/PHD/PHD-2001-01

[5] D. Augot and A. Zeh, "On the Roth and Ruckenstein Equations for the Guruswami-Sudan Algorithm," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008, pp. 2620–2624. [Online]. Available: http://dx.doi.org/10.1109/ISIT.2008.4595466

[6] G. Schmidt and V. R. Sidorenko, "Multi-sequence linear shift-register synthesis: The varying length case," in *Information Theory, 2006 IEEE International Symposium on*, 2006, pp. 1738–1742. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4036265

[7] G. Schmidt, V. R. Sidorenko, and M. Bossert, "Collaborative decoding of interleaved reed-solomon codes and concatenated code designs," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 2991–3012, 2009. [Online]. Available: http://dx.doi.org/10.1109/TIT.2009.2021308

[8] G. L. Feng and K. K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *Information Theory, IEEE Transactions on*, vol. 37, no. 5, pp. 1274–1287, 1991. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=133246

[9] G. Schmidt and V. R. Sidorenko, "Linear Shift-Register Synthesis for Multiple Sequences of Varying Length," May 2006. [Online]. Available: http://arxiv.org/abs/cs/0605044

[10] A. Zeh, C. Gentner, and D. Augot, "A Berlekamp-Massey Approach for the Guruswami-Sudan Decoding Algorithm for Reed-Solomon Codes," *Information Theory, IEEE Transactions on*, submitted for publication.

[11] E. R. Berlekamp, *Algebraic coding theory*. McGraw-Hill, 1968.

[12] R. J. Mceliece, "The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes," *Interplanetary Network Progress Report*, vol. 153, pp. 1–60, January 2003. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=2003IPNPR.153Q...1M

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach